UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/653,828	09/03/2003	Paul A. Martin	33226/972001; P7982	4116

32615 7590 12/13/2007

OSHA LIANG L.L.P./SUN
1221 MCKINNEY, SUITE 2800
HOUSTON, TX 77010

EXAMINER	
RAYYAN, SUSAN F	

ART UNIT	PAPER NUMBER
2167	

NOTIFICATION DATE	DELIVERY MODE
12/13/2007	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

lord@oshaliang.com
hernandez@oshaliang.com
DOCKETING@OSHALIANG.COM

Office Action Summary	Application No. 10/653,828	Applicant(s) MARTIN ET AL.	
	Examiner Susan F. Rayyan	Art Unit 2167	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 03 October 2007.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 14, 16-33 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 14 and 16-33 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-13, 15 are canceled. Claims 14, 16-33 are currently pending.

Claim Rejections - 35 USC § 101

2. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

the claimed invention is directed to non-statutory subject matter.

Regarding claims 21-26, claims are directed to software per se. A medium to store the software is needed. (for example, a computer readable storage medium encoding).

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 14, 16-33 are rejected under 35 U.S.C. 103(a) as being unpatentable over "Skip Lists: A Probabilistic Alternative to Balanced Trees" by William Pugh, herein after ("Pugh") in view of US Patent 6,651,146 B issued to Mysore Sathyanarayana Srinivas et al ("Srinivas").

As per claim 14 Pugh teaches:

instantiating nodes of the shared data structure in memory, wherein plural levels of same-direction referencing chains traverse respective subsets of the nodes in accordance with a key ordering relationship thereof, a first-level of the referencing chains traversing each node of the shared data structure and at least one other level of the referencing chains traversing less than all nodes of the shared data structure(Figure 1, element b includes a first level which points to each node (head->3->6->7->9) and Nth-level which points to a subset (head->6->9->19->);

wherein the insert-type operation performs a synchronized update of pointers beginning at the first level thereof and continuing upward (see Figure 1, insert operation and pointers);

wherein the delete-type operation performs a synchronized update of pointers beginning at a Kth level thereof and continuing downward to the first level (see page 2, Section Insertion and Deletion Algorithms, updating pointers).

Pugh does not explicitly teach operating on the shared data structure using insert-type and delete-type operations that are linearizable and lock-free for all concurrent executions thereof. Srinivas does teach this limitation (at column 8, lines 4-13, as adding and deleting data elements by one or more processors without the use of locks) to allow list management without the use of locks. It would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify Pugh with operating on the shared data structure using insert-type and delete-type operations that

are linearizable and lock-free for all concurrent executions thereof to allow list management without the use of locks (column 5, lines 57-67) as described by Srinivas.

As per claim 16 same as claim arguments above Pugh teaches:

wherein the insert-type operation performs a synchronized update of pointers in accordance with a first succession of the levels (see Figure 1, insert operation and pointers);

and wherein the delete-type operation performs a synchronized update of pointers in accordance with a second succession of the levels, the second succession opposing the first succession (see page 2, Section Insertion and Deletion Algorithms, updating pointers).

As per claim 17 same as claim arguments above Pugh teaches:

for a given one of the nodes instantiated, dynamically selecting a number of the plural, same-direction referencing chains that traverse the given node (Figure 2, Skip List search algorithm).

As per claim 18 same as claim arguments above Pugh teaches:

wherein the shared data structure implements a dictionary (page 1, Skip List Section, lines 1-2, dictionary).

As per claim 19 same as claim arguments above Pugh teaches:

wherein values are associated with respective ones of the keys, the method further as part of an execution of the insert-type operation introducing a value into the shared data structure (see Figure 3, node 17 is inserted);

and as part of an execution of the delete-type operation removing a value from the shared data structure the removed value corresponding to a search key (page 2, section Insertion and Deletion, delete type operation).

As per claim 20 same as claim arguments above Pugh teaches:

wherein the correspondence includes a greater-than-or-equal-to key match criterion (page 2, Figure 2, Skip List search algorithm, searchkey)

As per claim 21 Pugh teaches:

plural levels of same-direction referencing chains that traverse respective subsets of the nodes in accordance with a key ordering relationship thereof, a first of the referencing chains traversing each node of the shared data structure and a second of the referencing chains traversing less than all nodes of the shared data structure(Figure 1, element b includes a first level which points to each node (head->3->6->7->9) and Nth-level which points to a subset (head->6->9->19->);

wherein the insert-type operation performs a synchronized update of pointers beginning at the first level thereof and continuing upward (see Figure 1, insert operation and pointers);

wherein the delete-type operation performs a synchronized update of pointers beginning at a Kth level thereof and continuing downward to the first level (see page 2, Section Insertion and Deletion Algorithms, updating pointers).

Pugh does not teach plural nodes and a functional encoding of linearizable operations on the shared object, wherein the linearizable operations include both insert-type and remove-type operations and are lock-free for all concurrent executions thereof. Srinivas does teach this limitation (at column 8, lines 4-13, as adding and deleting data elements by one or more processors without the use of locks) to allow list management without the use of locks. It would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify Pugh with plural nodes and a functional encoding of linearizable operations on the shared object, wherein the linearizable operations include both insert-type and remove-type operations and are lock-free for all concurrent executions thereof to allow list management without the use of locks (column 5, lines 57-67) as described by Srinivas.

As per claim 22 same as claim arguments above Pugh teaches:

wherein, on at least some executions, the insert-type introduces an additional node into at least one of the referencing chains (see Figure 3, node 17 is inserted).

As per claim 23 same as claim arguments above Pugh teaches:

wherein, on at least some executions, the delete-type operation excises a particular node from all referencing chains that traverse the particular node (lines 21-36 of Figure 4 and page 2, Section Insertion and Deletion, delete type operation).

As per claim 24 same as claim arguments above Pugh teaches:

wherein the delete-type operation employs a greater-than-or-equal-to key match criterion (lines 1-4 of Figure 4, delete operation).

As per claim 25 same as claim arguments above Pugh teaches:

wherein the shared object implements a shared skip-list-type data structure (page 1, Section Skip List, skip list data structure).

As per claim 26 same as claim arguments above Pugh teaches:

wherein the shared object implements a dictionary data structure (page 1, Skip List Algorithm, dictionary structure).

As per claim 27 Pugh teaches:

In a computational system that employs a shared list-type data structure that includes plural nodes and plural levels of referencing chains that traverse respective ones of the nodes in accordance with an ordering thereof, wherein a higher-level one of the referencing chains traverses no more than a subset of the nodes traversed by a lower-level one of the referencing chains(Figure 1, element b includes a first level which points to each node (head->3->6->7->9) and Nth-level which points to a subset (head->6->9->19->);

deleting a node from the shared list-type data structure by excising the node from successive ones of the referencing chains, beginning with a highest-level one of the referencing chains that traverses the node and continuing through a lowest-level one of the referencing chains, wherein each such excision employs a linearizable synchronization operation to bridge the excised node and associate a dead pointer indication therewith (lines 21-38 of Figure 4, delete node and Section Insertion and Deletion Algorithm of page 2,update contains pointer) ;

inserting a node into the shared list-type data structure by introducing the inserted node into one or more of the referencing chains, beginning with the lowest-level referencing chains and continuing through successive zero or more higher-level referencing chains (Figure 3, Inserting node).

Pugh does not explicitly teach facilitating lock-free concurrent operations on the shared list-type data structure. Srinivas does teach this limitation (at column 8, lines 4-13, as adding and deleting data elements by one or more processors without the use of locks) to allow list management without the use of locks. It would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify Pugh with facilitating lock-free concurrent operations on the shared list-type data structure to allow list management without the use of locks (column 5, lines 57-67) as described by Srinivas.

As per claim 28 same as claim arguments above Srinivas teaches:

wherein all concurrent executions of the deleting and the inserting are linearizable and lock-free at column 8, lines 4-13, as adding and deleting data elements by one or more processors without the use of locks.

As per claim 29 same as claim arguments above Pugh teaches:

wherein the dead pointer indication includes one: a self pointer, a pointer to a dead node, a back pointer (page 2, Section Insertion and Deletion lines 2-7, update contains the pointer and Figures 3-4).

As per independent claim 30 Pugh teaches:

defining a shared list-type data structure that includes plural nodes and plural levels of same direction referencing chains that traverse at least some of the nodes in accordance with an ordering thereof, wherein a higher-level one of the referencing chains traverses no more than a subset of the nodes traversed by a lower-level one of the referencing chains(Figure 1, element b includes a first level which points to each node (head->3->6->7->9) and Nth-level which points to a subset (head->6->9->19->);

wherein the inserting comprises performing a synchronized update of pointers beginning at the first level thereof and continuing upward (see Figure 1, insert operation and pointers);

wherein the deleting comprises performing a synchronized update of pointers

beginning at a Kth level thereof and continuing downward to the first level (see page 2, Section Insertion and Deletion Algorithms, updating pointers).

Pugh does not explicitly teach inserting into, and deleting from, the shared list-type data structure, wherein all concurrent executions of the deleting and the inserting are linearizable and lock-free. Srinivas does teach this limitation (at column 8, lines 4-13, as adding and deleting data elements by one or more processors without the use of locks) to allow list management without the use of locks. It would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify Pugh with inserting into, and deleting from, the shared list-type data structure, wherein all concurrent executions of the deleting and the inserting are linearizable and lock-free to allow list management without the use of locks (column 5, lines 57-67) as described by Srinivas.

As per claim 31 same as claim arguments above Pugh teaches:

a definition of a skip list instantiable in storage (page 1, section Skip Lists and Figure 1, as skip list data structures and page and page 2 section Insertion and Deletion Algorithms, as insert and delete nodes).

Pugh does not explicitly teach lock-free means for coordinating concurrent and linearizable executions of both insert-type and delete-type operations. Srinivas does teach this limitation (at column 8, lines 4-13, as adding and deleting data elements by one or more processors without the use of locks) to allow list management without the

use of locks. It would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify Pugh with lock-free means for coordinating concurrent and linearizable executions of both insert-type and delete-type operations to allow list management without the use of locks (column 5, lines 57-67) as described by Srinivas.

As per claim 32 same as claim arguments above Pugh teaches: the storage (Figure 1 and section "Additional Work on Skip Lists", paragraph 1 as multiple processors and shared memory).

As per claim 33 same as claim arguments above Pugh teaches:
plural processors, the insert-type and delete-type operations executable (Figure 1 and section "Additional Work on Skip Lists", paragraph 1 as multiple processors and shared memory).

Response to Arguments

4. Applicant's arguments filed October 3, 2007 have been fully considered but they are not persuasive.

Applicant argues Pugh does not teach insert-type operations and delete-type operations that update pointers in opposite directions. Pugh does teach wherein the insert-type operation performs a synchronized update of pointers beginning at the first level thereof and continuing upward (see Figure 1, insert operation and pointers) and

wherein the delete-type operation performs a synchronized update of pointers beginning at a Kth level thereof and continuing downward to the first level (for kth level equals first level see page 2, Section Insertion and Deletion Algorithms, updating pointers).

Applicant argues Pugh does not teach locking and can not be combined with Srinivas which teaches lock-free updates. Examiner finds Applicant indicated in the specification on page 3, paragraph 7, that Pughs' technique allows concurrent operations to execute. Srinvas teaches insert and delete operations which are lock-free (at column 8, lines 4-13, as adding and deleting data elements by one or more processors without the use of locks). It would have been obvious to modify the cited references to allow list management without the use of locks (column 5, lines 57-67) as described by Srinivas.

Contact Information

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Susan F. Rayyan whose telephone number is 571-272-1675. The examiner can normally be reached on M-F, 7:30-4:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cottingham can be reached on 571-272-7079. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Susan Rayyan
December 7,2007

JOHN COTTINGHAM
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100